Unsupervised Scene Detection using β-VAE

Zhe Chen
Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
zhechen2@andrew.cmu.edu

Yinghao Ma
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
yinghaom@andrew.cmu.edu

Ruiyang Jin
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
ruiyangj@andrew.cmu.edu

Fan Zhou
Electrical & Computer Eng.
Carnegie Mellon University
Moffett Field, CA 94035
fanzhou@andrew.cmu.edu

Abstract

Scene changes are a bifurcation between image representations, with characters, scenery, objects, or perspectives rapidly changing. An ideal scene change detection solution is unsupervised, averting the need for labeling images. We create such an unsupervised approach with β-VAEs and proposed a few architectural modifications. We demonstrate greater than 80% top K accuracy given K scenes using a tuned β-VAE model. We also conduct ablation studies on the experiment results with their effects explained and provide additional discussion on potential future works.

1 Introduction

Automatic scene detection task focuses on grouping the frames in a video into different scenes using computer algorithms, and is becoming one of the most valued feature in video creation and consumption. Content creators can rely on this technology to quickly find their desired scene in a long clip. Content consumers can enjoy better video tagging and metadata searching without additional human effort.

Earlier work in this domain relies on threshold-based algorithms [11] and recently, Rao etc. [10] introduced a supervised multi-modal model for scene detection. However, while unsupervised learning can help us finish this task without labeling image, currently there is no attempt to approach this challenge using unsupervised learning. Therefore, we propose to use β-Variational Autoencoder (β-VAE) and its variants to detect scene changes in an unsupervised learning manner. We use Convolutional Neural Network (CNN) to encode and decode frames, and then examine the β-VAE's latent space to extract the latent features relevant to scene changes, which are used in different ways for detection.

In this project, we use the term "semantic scene change" to indicate meaningful scene changes in terms of story-telling as some scene changes can be trivial (e.g. changes from/to a black scene). Figure 1 shows such an example, as the frame pairs, although similar, are from two different semantic scenes, that are the protagonists' respective bedrooms from the anime movie *Kimi No Na Wa*.

The TA mentor for this project is Joseph Konan. Some of his materials are included in our paper.

Preprint. Under review.

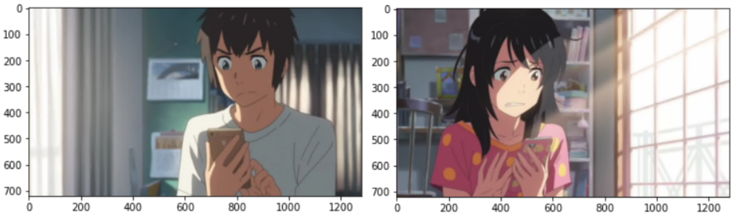


Figure 1: Semantic Scene Change

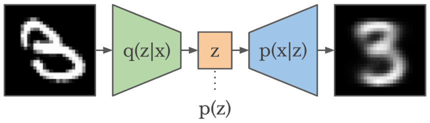


Figure 2: Schematic diagram of VAE

2 Literature Review

Comparing to supervised learning, unsupervised learning can learn to detect scene changes using unlabeled data[6]. This is potentially more cost-effective as manual labelling is time-consuming.

As early as 2001, Shu-Ching Chen et al. had used unsupervised methods to detect scene changes in video scenes[5]. They combined unsupervised segmentation algorithm and object tracking to extract the segmentation mask map of each video frame. The frames are compared using the difference between their segmentation mask maps. However, most of the early techniques rely on the extraction of low level features, like pixel value or color histograms[1]. More recently, the use of unsupervised high-dimensional feature extractor in the scene recognition task, such as latent Dirichlet allocation (LDA) and multivariate alteration detection (MAD) is proven to have better performance[6].

As the application of neural networks such as CNN, VAE[8] and ALBERT [9] becomes ubiquitous and outperforms traditional models in a wide variety of fields, it's possible to detect scenes more accurately using deep learning. Christopher P. Burgess et al. proposed Multi-Object Network (MONet)[3], which is capable of learning to decompose and represent challenging 3D scenes into semantically meaningful components. In MONet, a VAE is trained end-to-end together with a recurrent attention network — in a purely unsupervised manner -– to provide attention masks around, and reconstructions of, regions of images.

VAE is a framework designed to learn the joint distribution of the actual image X and a set of generative latent factors Z which is independent and interpretable. And the inferred posterior configurations of the latent factors Z is believed to be a normal distribution $q_\phi\left(z|x\right) := \mathbb{P}\left(Z|X\right)$ with parameter ϕ. We use the KL-divergence to make $q_\phi\left(z|x\right)$ close to the standard normal distribution for better disentangled manner [8]. Besides, Z should hold the ability to generate the observed data X with parameter θ. We denote the probability as $p_\theta(x|z)$. The optimum of loss function should maximum its log-likelihood.

β-VAE uses a refined loss function with $\beta > 1$ [7] and the VAE becomes a special case with $\beta = 1$.

$$L = -\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + \beta D_{KL}\left(q_\phi\left(z|x\right)||\mathcal{N}\left(0, I\right)\right)$$

With information theory techniques, Chen et al. [4] further separated β-VAE's loss function in to three terms: index-code mutual information, total correlation and dimension-wise KL. They applied β only on the total correlation term and found comparable or better results, known as β-TCVAE.

3 Models

3.1 Baseline

The traditional baseline model is a simple dynamic threshold over the L2 norm of a frame sequence. In practice, we found that calculating the L2 norm of per-pixel differences between adjacent frame pairs to be a much more comparable way to that of the other models.

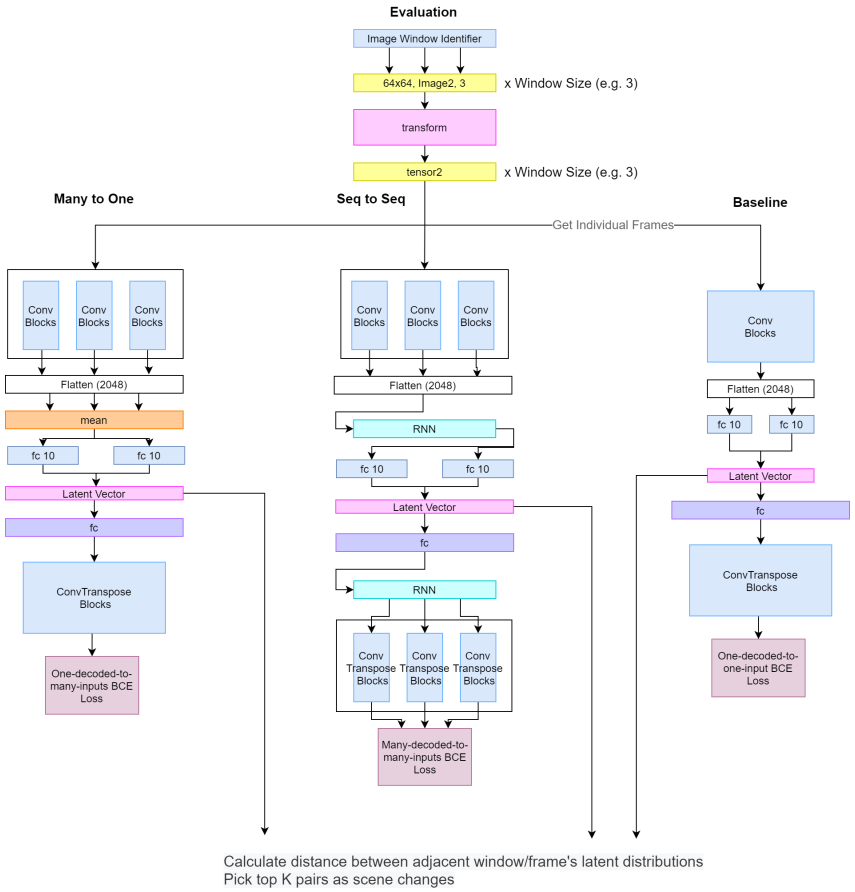


Figure 3: Models Summary

The β-VAE baseline model architecture would be the original architecture trained on every individual frames. We will use the open-source version [12].

3.2 Architectural Search

"Many to One": while training, we can feed the model with a window of consecutive frames instead of a single frame, potentially incorporating information about a short period of time instead of one single frame. To implement this modification, we still use CNN as encoder, and input a window of frames into CNN at each iteration and transform the mean of the features embedding across a window into latent dimensions. To train such a model, the single decoded image is compared against all input images as we assume that the decoded image should contain the features from all input frame.

"Seq to Seq": alternatively, we can use a Convolutional Recurrent Neural Network (CRNN) architecture as encoder, in which the CNN outputs one result for each frame in the window, and an RNN encoder processes the embedding sequence. The last output of the RNN encoder is used to calculate the latent distributions. For reconstruction, we can either use a RNN decoder to decode the sample z back into a sequence of the images, or use the same decoder as the "Many to One" model.

Figure 3 summarizes how these models are to be trained and evaluated.

3.3 Refined Loss

We proposed three kind of loss function for the β-VAE. The original β VAE multiply the value of KL divergence with the parameter β to increase the penalty of the latent vector being too concrete with small variance, thus increase the capacity of disentanglement. We called it "H" loss.

Our second loss function is a refined loss and is believed to be able further preventing the KL divergence from being too large [2]

$$L = -\mathbb{E}_{z \sim q_{\phi}(z|x)} \left[\log p_{\theta}(x|z) \right] + \gamma \left| D_{KL} \left(q_{\phi} \left(z|x \right) || \mathcal{N} \left(0, I \right) \right) - C \right| \tag{1}$$

Here, the KL divergence will not become to negative infinity too fast an might be limited around the mildly increasing parameter C.

Our third loss function is to use the Wasserstein distance between the standard normal distribution and the latent to replace the (refined) KL-divergence.

$$L = -\mathbb{E}_{z \sim q_{\phi}(z|x)} \left[\log p_{\theta}(x|z) \right] + \beta \times d_{\text{Wasserstein}} \left(q_{\phi} \left(z|x \right), \mathcal{N} \left(0, I \right) \right) \tag{2}$$

3.4 Evaluating Changes in Latent Vector

The model β-VAE use its value change in latent vector to evaluate the scene change in the original images. While the latent vector is actually a normal distribution instead of a vector and several new approaches to evaluate the latent vector difference has been used.

First, we can evaluate the KL-divergence between the two latent distributions. Due to the incommutable of the KLD, there are two ways to calculate the KL-divergence with different order.

Beside KL-divergence, the Wasserstein distance is by nature a outstanding metrics between different distribution. Here, the distance is actually determined by the concatenation of means μ and variance σ^2 for the latent distribution — the Wasserstein distance between two independent Gaussian distribution is shown below:

$$\begin{aligned} d_{\text{Wasserstein}} \left(\mathcal{N}_1, \mathcal{N}_2 \right) &= \left\| \mu_1 - \mu_2 \right\|^2 + trace \left[\Sigma_1 + \Sigma_2 - 2 \left(\Sigma_1^{0.5} \Sigma_2 \Sigma_1^{0.5} \right) \right] \\ &= L_2 \left((\mu_1, \sigma_1), (\mu_2, \sigma_2) \right) \end{aligned} \tag{3}$$

where μ_1 and μ_2 are the mean of two distribution, and σ_1 and σ_2 are their variance respectively.

What's more, some of the methods previously proposed in mid-term report has not been implemented for the following reasons. In unsupervised tasks, whether the semantics inside a window include scene change instead of others is hard to evaluate. As a result, we abandon the methods that use the latent vector or latent distribution itself for ranking or classification. Furthermore, From the experiments shown in the later section, we could find that the better reconstruction may leads to worse results on scene detection since the latent vector keeps too much details other than scene semantics. As a consequence, we abandon the indicator of reconstruction results.

4 Experimental Evaluation

4.1 Dataset and Test Setup

We use all frames from the movie *Kimi no Na wa* at 720p resolution. We pre-converted all images to 256p, which greatly alleviated the CPU bottleneck in loading the images. The images are further resized with adaptive average pooling to make our model more computationally efficient. The dataset contains 191881 frames in total. The leading 31136 frames (≈16.7 minutes) are reserved as the validation set and the rest are used as training set.

We also used an 142p version of the movie with only unique frames for validation. Our TA Konan manually labelled the scene changes in the leading ≈16.7 minutes of this dataset. Due to unforeseen misalignment, we are unable to automatically map these labels to the 720p dataset. Therefore, we have to use this version of dataset for validation instead.

Kimi no Na wa was originally produced by CoMix Wave Films and directed by Makoto Shinkai. We understand that this dataset is copyrighted. We will only use this dataset in accordance with Fair Use [17 U.S. Code § 107]. Specifically, we are only using this dataset for teaching, scholarship, and research, which is explicitly allowed under Fair Use.

All our models are trained on an AWS `g4dn.xlarge` instance with 4 vCores, 16GB RAM and a Nvidia T4 GPU with 16GB VRAM. The spot instance of this type costs $0.16 per hour. Per epoch time is 3 minutes for models using 128×128 images and 2 minutes for models using 64×64 images.

4.2 Evaluation Metrics and Aggregated Results

As mentioned in section 3.4, We use element-wise L2 norm of 2 jointly statistics (μ and σ^2), that is, Wasserstein distance of two independent Gaussian distributions to compare the adjacent latent vectors for the β-VAE, while the traditional baseline model directly uses the raw normalized image pixel values as features to evaluate the L2 distance among pictures.

There are 257 scene changes in the validation data. Thus, we picked 257 adjacent frame pairs with the largest Wasserstein distance as our model prediction. Then we evaluate the accuracy on those 257 images using the labelled scene changes. Based on the results of our ablation study in section 4.3, we used the following settings for our final models:

- 64×64 input images
- 10 latent dimensions using B loss
- Adam optimizer with a learning rate of 10^{-3}
- ReduceLROnPlateau scheduler with relative threshold of 0.001 and patience of 3 epochs.

Model Type	Hyperparameters	Accuracy
Traditional	-	0.790
Baseline β-VAE	$\gamma = 750$, epoch 16	0.828
Many-to-one β-VAE	$\gamma = 1000$, epoch 40	0.798
Seq-to-seq β-VAE	$\gamma = 1000$, epoch 38	0.400

Table 1: Highest-performing Models Comparison

Table 1 shows the highest-performing models of each architectural type and their achieved validation accuracies. Clearly, our proposed architectural modifications didn't result in any improvements as they might need more epoch to converge especially RNN structure, but nevertheless, our ablation study has provided a good set of hyperparameters for the baseline model to achieve respectable performance to start with.

4.3 Ablation Study

For this study, we mainly used a baseline architecture with 128×128-sized images, and B-type loss with $\gamma = 1000$, max capacity $= 25$, iteration $= 10^5$ and is trained for 20 to 40 epochs using an learning rate of 10^{-3}.

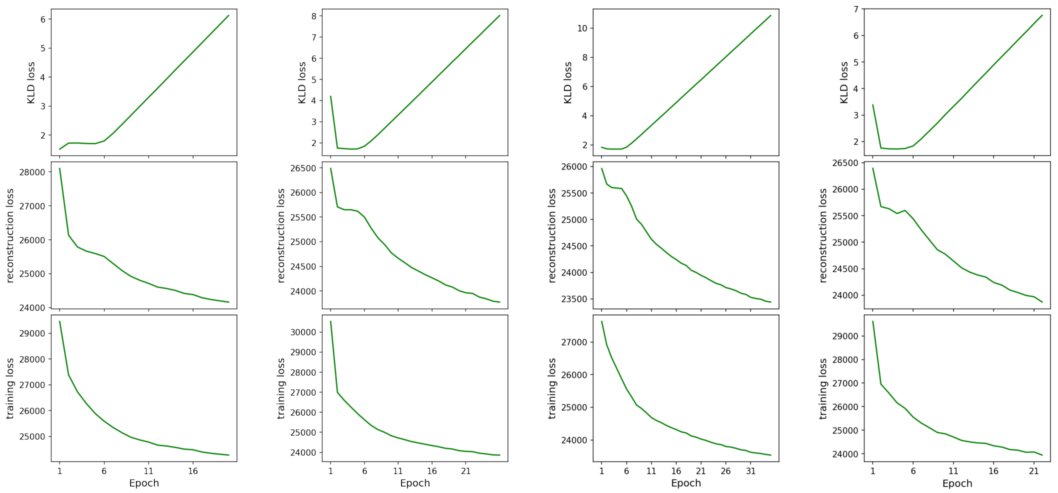
4.3.1 Optimizer

We adopted a hyperparameter regulation method similar to the Grid Search, and used the baseline model to find the optimal training hyperparameter for our optimization model and the results are shown in figure 4. First, we divided the training into three groups according to the candidate SGD, Adam and AdamW, and searched for the optimal learning rate of each optimizer.

When using SGD optimizer, the reconstruction loss and the KLD loss become NaN since the first epoch with a learning rate $\geq 10^{-4}$, and the situation doesn't get better after the momentum is turned off. The reason might be that the logvar layer is not initialized properly and leads to a blow up in following values and gradients. After adding gradient clipping and using zero initialization for logvar layer, the model is able to learn with a learning rate of 0.01 using SGD.

For Adam and AdamW optimizers, using a learning rate $\geq 10^{-2}$ or higher results in either the loss blowing up or becoming NaN, so gradient descent could not be carried out, while 1e-3 training had better effect.

This suggests that the β-VAE model is quite sensitive to gradient updates and can only be optimized using relatively small learning rate and an adaptive momentum optimizer such as Adam. Based on the loss characteristics below, we concluded that Adam works best for our model. Therefore, all our subsequent models are trained with the Adam optimizer.



(a) SGD Optimizer (ReduceLROnPlateau)

(b) AdamW Optimizer (ReduceLROnPlateau)

(c) Adam Optimizer (ReduceLROnPlateau)

(d) Adam Optimizer (CosineAnnealingLR)

Figure 4: Effect of Optimizer and LR Scheduler on Model Training

4.3.2 Scheduler

As for the selection of scheduler, there is no significant differences between our candidate schemes: ReduceLROnPlateau and CosineAnnealingLR, as the decreasing trends of loss shown in Figure 4 (c) and (d) do not differ significantly. We decided to use a the familiar and controllable ReduceLROnPlateau scheduler.

After selecting the ReduceLROnPlateau, we tuned the main hyperparameters using Grid Search, and finally determined the optimal ones: $\text{factor} = 0.5, \text{patience} = 3, \text{threshold} = 10^{-3}$

4.3.3 Loss function

Here are the results of three different loss functions we used in the project in Figure 5: Original loss H, refined loss B and the loss with Wasserstein distance (W loss). Our experiments shows that this refined B loss function can improve the accuracy of scene detection although it has higher loss on the maximum likelihood estimation term and gives worse reconstruction results. Note that our W loss graph is trained using 128 latent dimensions instead of 32 latent dimensions, however, our later comparison also shows that it doesn't improve the baseline model's performance.

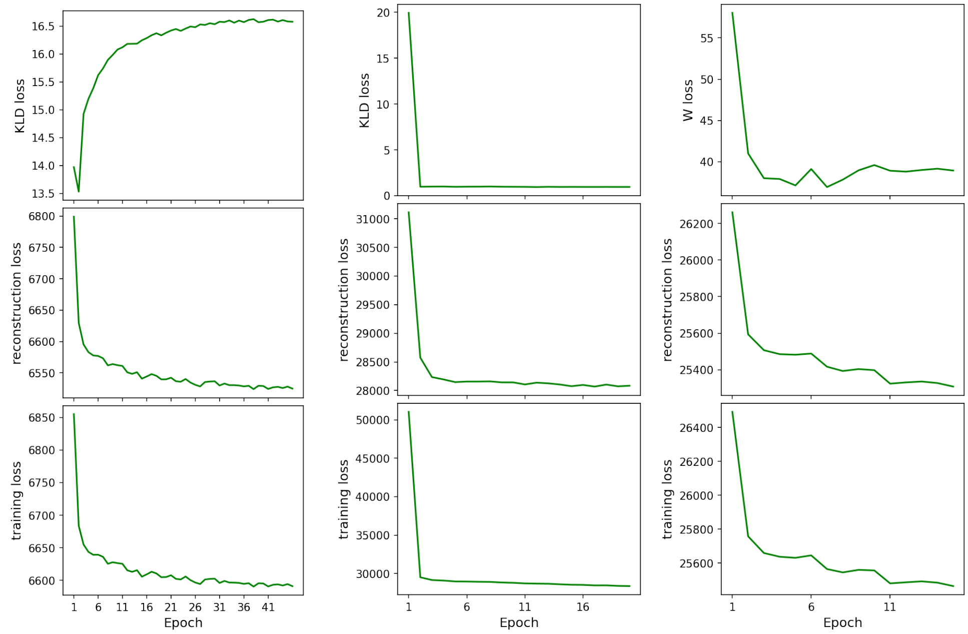
The original H loss, though keeping better details in the decoded image, does not provide better scene semantic disentanglement. We think this is caused by the higher KL divergence when trained with the original loss. When Wasserstein distance is used to replace the KL divergence, we do not see any improvement in detection accuracy.

We also studied the effect of hyperparameters of the original H loss and refined B loss and tabulated the result in Table 2. We concluded that by forcing a bound on the latent distribution's KL divergence, the disentanglement of scene semantics is more effective. However, too much penalization would also limit the expressiveness of the latent distributions and making the model ineffective in image decoding and scene detection. Hence, we decided to use the refined B loss with $\gamma = 1000$ or $\gamma = 750$ for our later experiments.

	$\gamma = 10$	$\gamma = 50$	$\gamma = 100$	$\gamma = 200$	$\gamma = 500$	$\gamma = 750$	$\gamma = 1000$
Accuracy	0.790	0.794	0.805	0.813	0.825	0.828	0.809

	$\beta = 2.5$	$\beta = 4$	$\beta = 10$	$\beta = 20$
Accuracy	0.801	0.817	0.805	0.798

Table 2: Loss Parameter Comparison



(a) Original H Loss

(b) Refined B Loss

(c) W Loss

Figure 5: Effect of Different Types of Loss

4.3.4 Network Architecture

As tabulated in Table 1, our modified architectures didn't yield any performance improvements over the carefully tuned baseline.

Since the 142p validation dataset only contains unique frames, we fed our many-to-one model with individual frames as the original sliding window approach uses consecutive, rather than unique frames. As for the seq-to-seq model, we only fed the model with a sliding window of 2 adjacent frames to be consistent with the training data.

In many-to-one model, we calculate the reconstruction loss between the mean of the images in a window and the reconstructed image. Since the whole window is aggregated in one single latent variable, its information about image features are blurred by the contexts to some degree. When comparing two latent variables to detect scene change, those blurred latent variables may not be distinct enough to represent a scene change, leading to possible performance degradation. In this case, it might be better to rank the individual latent variables instead of calculate the distance between latent variables, since the latent variable already has information of a whole window. Also, taking the mean after flatten layer may not be the best way to utilize context information.

For the seq-to-seq model, We first used the proposed RNN decoder using Elman RNN. This is because our TA Konan instructed us that Elman RNN is the proper transpose operation of itself. However, the network cannot be trained to reconstruct images at all. We hypothesised that the information bottleneck from the latent dimension and RNN's hidden output is too severe to enable effective decoding.

Trying to revamp the seq-to-seq model, we replaced the model decoder with that of the Many-to-One model and Elman RNN with LSTM. Though this model can be trained, its accuracy is much worse than the other 2 models and the reconstructed images are quite bad. This might be caused by slow convergence of LSTM-based model, but the exact cause is not yet certain.

4.3.5 Latent Dimension Size

We carried out experiments on the effect of varying latent dimension size. For the same many-to-one model, using 10 latent dimensions yields an final accuracy of 0.805, as compared to 0.747 with 32 latent dimensions and 0.669 with 128 latent dimension. As such, small latent dimension (=10) is enough to embed scene information and make a good quality disentanglement on scene semantics.

Increasing latent dimension size actually decreased the accuracy of scene detection, although it includes much more information and usually leads to better reconstructions. We hypothesised that this is because the additional latent dimensions starts to focus on the scene details rather than the overall scene semantics, leading to worse scene detection accuracy.

5 Conclusion

Automatic scene detection is an important technique in computer vision, but there are current no unsupervised deep-learning models to perform this task. In this project, we proposed to use β-VAE to address this problem. We first examined the effect of different hyperparameter combinations to the performance of β-VAE by an ablation study and found a better set of hyperparameters. We then compared the accuracy of β-VAE and its variants to the traditional baseline on scene change detection task. β-VAE could achieve relatively high accuracy on our dataset with proper hyperparameters, but our proposed modifications don't necessarily improve accuracy further. After analyzing the model and the results, we gained a deeper understanding to the principle of β-VAE, and honed our skill in training and evaluating models.

6 Discussion and Future Works

6.1 L2 Baseline is not Good Enough

Although the β-VAE does not improve scene detection accuracy a lot as compared to the traditional L2 difference ranking, we observed that the β-VAE baseline model is able to identify more semantic scene changes and less trivial scene changes. For this comparison, we picked top 100 divergent frame pairs in L2 baseline and an H-loss β-VAE, and manually labelled the types of each detected scene changes below in Table 3.

The results of L2 baseline are mostly frame pairs with stark brightness differences as the L2 norm would be bigger for frame pairs of larger differences in pixel values. In contrast, the β-VAE baseline model has learned to extract some semantic meanings from the frames, although not performing perfectly as we desired. Figure 6 illustrated a camera zooming case flagged by the β-VAE model. This cannot be detected using the traditional model due to comparable image brightness. Therefore, the prediction of β-VAE models is much more meaningful.

Model	SE	BL	AN	ZO	BR	NO
Traditional	43	22	8	7	8	12
Baseline β-VAE	52	10	10	14	4	10

Table 3: Baseline Model Top 100 Divergent Frame Pair Labels

Notations: SE - semantic scene change, BL: change to/from black scene, AN: camera angle change, ZO - zoom in/out, BR - brightness change, NO - not a scene change

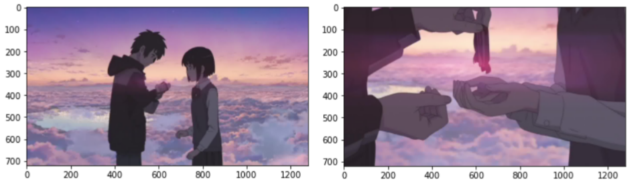


Figure 6: Camera Zooming in on a Semantic Scene

6.2 Major Challenges and Project Constraints

Compared with our homework, the data we used in this project comes from a real movie, presented to us without much preprocessing. Therefore, we must choose appropriate methods to preprocess the data before getting into scene detection task. For example, we need to choose the resolution of the input image so that our training is both fast and efficient, which is as important as choosing hyperparameters. Besides, compared with artifact datasets, it's hard to gain better performance on real world data, since it's not designed for the task, and we need to define scene changes, which is confusing even to humans.

Another challenge comes from the model architecture. β-VAE is more complicated than typical models like CNN and RNN, and its theoretical basis involves a lot of math. So it's harder to propose efficient modifications with solid mathematical reasoning, and we have to rely on heuristics and try many kinds of modifications, which could be fruitless if we happen to enter a wrong way. We've tested many combinations of hyperparameters and some variants, but none of them makes a breakthrough in accuracy.

6.3 Possible Future Work

The train loss is a certain kind of β-VAE which shows better results, while in many cases, β-TCVAE shows best result and this deserves more attention.

Besides, in many-to-one method, the accuracy is likely to become higher if we change the input from the mean of images in a window to their variance. The intuition behind this could be that we can form an input window by not only adjacent images but their statistics, and use β-VAE to retrieve other useful features from them. A possible improvement would be using feeding the model with statistics of image windows, and find the input that yields best result.

Also, our seq-to-seq β-VAE doesn't get expected result and we haven't fully understood the reason. One of the possible future improvements could be exploring how to incorporate RNN more effectively.

7 Appendix

Our work could be found at GitHub, and the link is `https://github.com/Chen-Zhe/team-two-spiked-apes`.

Most of our work, including literature review, model design, and many other parts, are done by the whole team together, either in the form of team discussion or distributing the work equally to each person. When doing experiments, Zhe Chen set up notebooks for running baseline model, plotting the result and evaluating models, and all four members run different experiments based on them.

References

[1] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2015. Shot and Scene Detection via Hierarchical Clustering for Re-using Broadcast Video. In *Computer Analysis of Images and Patterns*, George Azzopardi and Nicolai Petkov (Eds.). Springer International Publishing, Cham, 801–811.

[2] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, and A. Lerchner. 2018. Understanding disentangling in β-VAE. (2018).

[3] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. 2019. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390* (2019).

[4] Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating Sources of Disentanglement in Variational Autoencoders. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. `https://proceedings.neurips.cc/paper/2018/file/1ee3dfcd8a0645a25a35977997223d22-Paper.pdf`

[5] Shu-Ching Chen, Mei-Ling Shyu, Cheng-Cui Zhang, and R.L. Kashyap. 2001. Video scene change detection method using unsupervised segmentation and object tracking. In *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.* 56–59. `https://doi.org/10.1109/ICME.2001.1237654`

[6] B. Du, Y. Wang, C. Wu, and L. Zhang. 2018. Unsupervised Scene Change Detection via Latent Dirichlet Allocation and Multivariate Alteration Detection. *IEEE Journal of Selected*

Topics in Applied Earth Observations and Remote Sensing 11, 12 (2018), 4676–4689. https://doi.org/10.1109/JSTARS.2018.2869549

[7] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. Beta-VAE: Learning basic visual concepts with a constrained variational framework. (2016).

[8] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

[9] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations, In International Conference on Learning Representations, ICLR 2020. *arXiv preprint arXiv:1909.11942*.

[10] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. A Local-to-Global Approach to Multi-modal Movie Scene Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10146–10155.

[11] Giorgio Rascioni, Susanna Spinsante, and Ennio Gambi. 2010. An Optimized Dynamic Scene Change Detection Algorithm for H.264/AVC Encoded Video Sequences. *International Journal of Digital Multimedia Broadcasting* 2010 (2010), 1–9. https://doi.org/10.1155/2010/864123

[12] A.K Subramanian. 2020. PyTorch-VAE. https://github.com/AntixK/PyTorch-VAE. (2020).